Fault Tolerant Approaches based on Evolvable Hardware and using Reconfigurable Electronic Devices

Didier Keymeulen, Adrian Stoica, Ricardo Zebulum, Yili Jin and Vu Duong
Center for Integrated Space Microsystems
Jet Propulsion Laboratory
California Institute of Technology

http://cism.jpl.nasa.gov/ehw/darpa

Presentation to the IEEE International Integrated Reliability Workshop (IRW-2000)
October 23-26, 2000

This work was sponsored by the
National Aeronautics and Space Administration and
Defense Advanced Research projects Agency

Presentation Outline

1. Introduction: Survivability and Versatility.
2. EHW: Reconfiguration Mechanism and Reconfigurable Hardware
3. Circuits Synthesized by Extrinsic and Intrinsic Evolution
4. Evolutionary Approaches to Fault Tolerance
5. Fault Tolerant Experiments
6. Conclusion

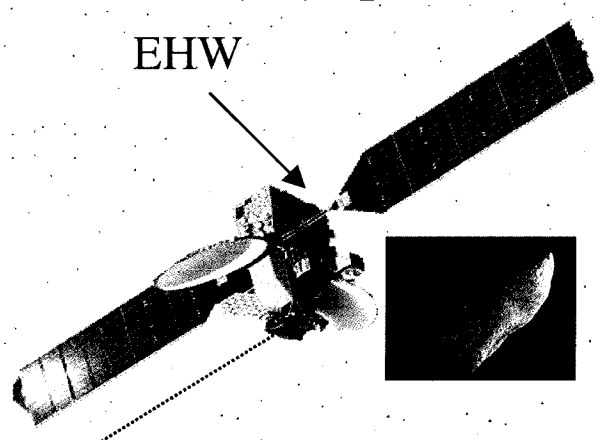
Survivability & Versatility

- new environments, new missions, coping with malfunctions -

Survivability: Maintain functionality through parametric adjustments even with changes in hardware characteristics (self-healing design)

- Temperature variations
- Radiation impacts
- Aging
- Malfunctions, etc.

Versatility: Create new functionality through synthesis of a totally new circuits for dramatic changes in hardware/environment (optimal design)

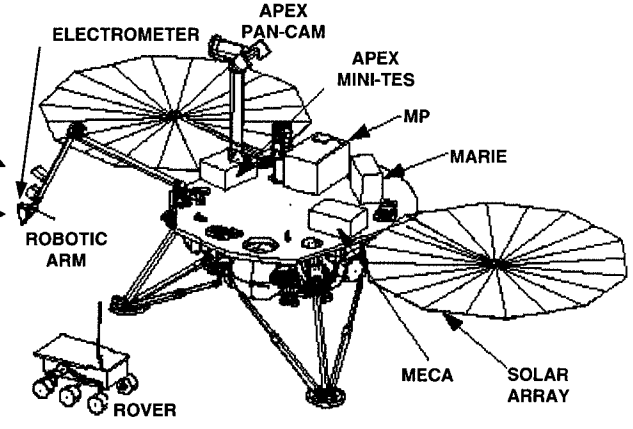


New functions required for:

- New mission phases
- Missions where findings require re-scooping to take advantages of new opportunities

Up-link new functions for re-planned mission

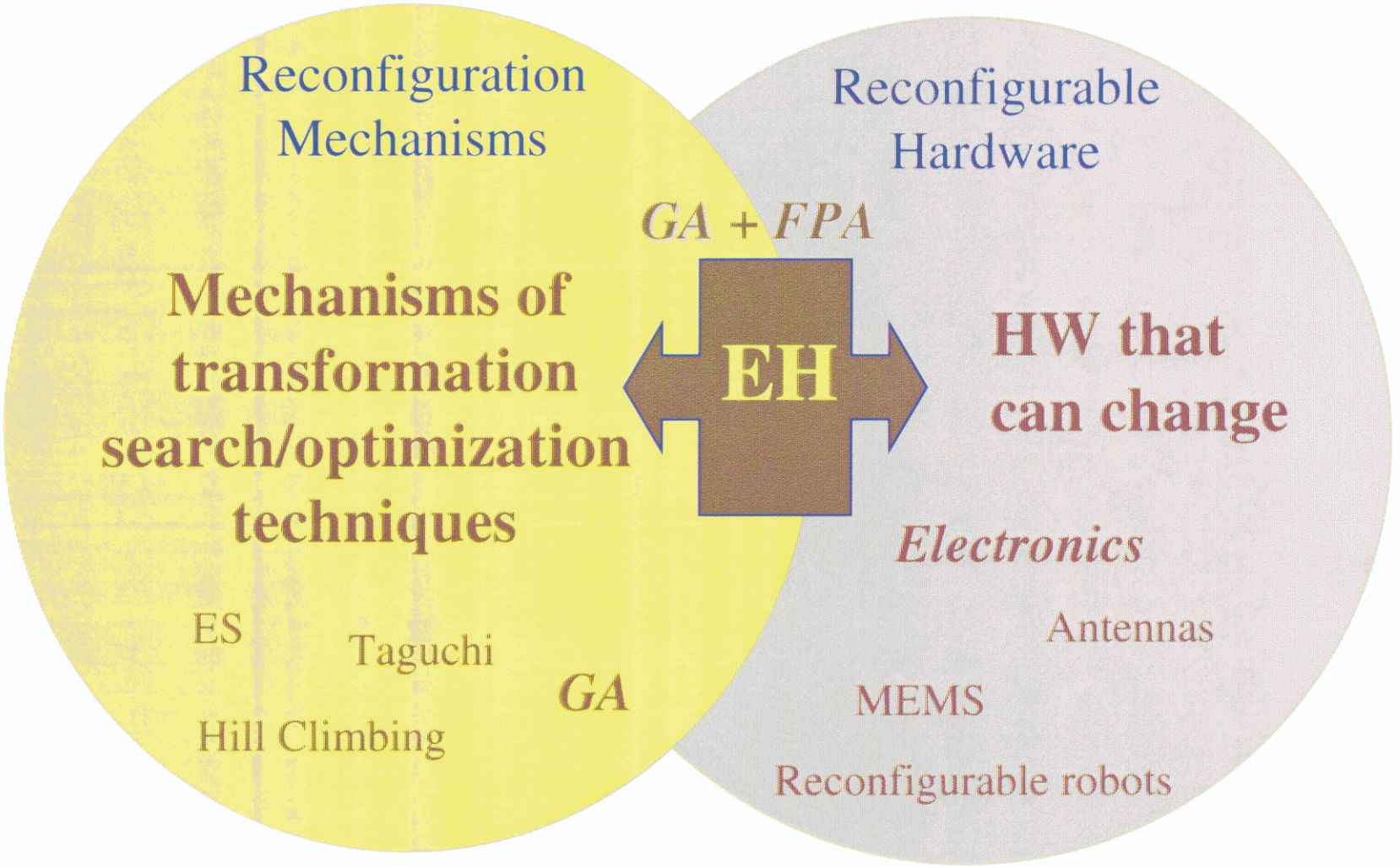
EHW



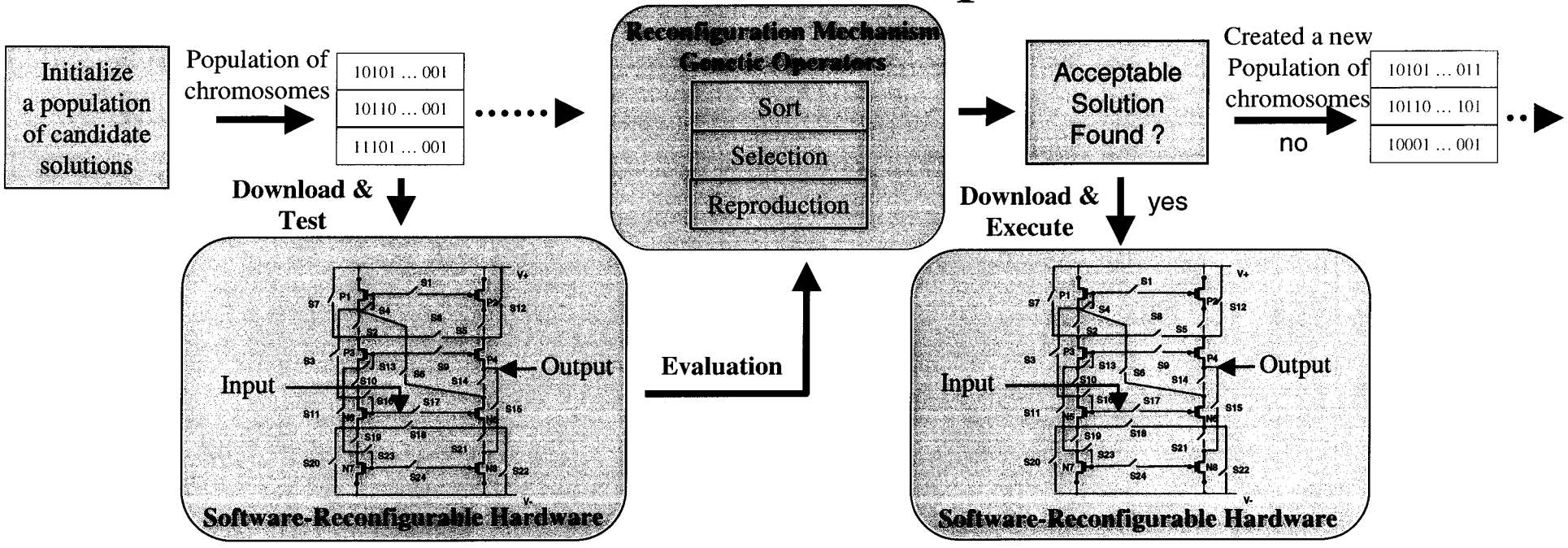
Accurate model of hardware is not available after launch

What is EHW ?

Evolvable Hardware =
Reconfiguration Mechanism + Reconfigurable Hardware



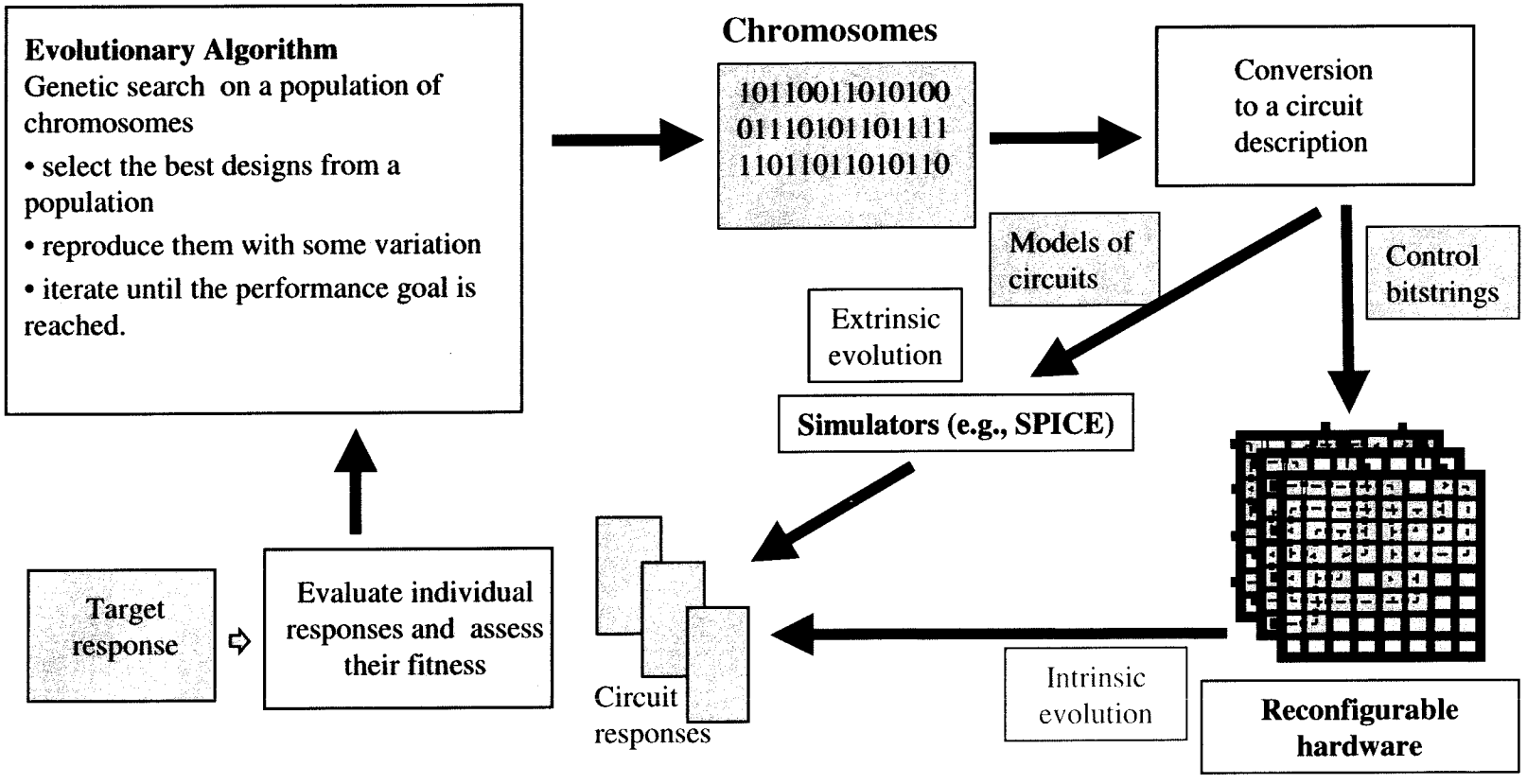
Evolvable Hardware Implementation



- The idea behind evolutionary synthesis/adaptation of electronic circuits is to employ a genetic algorithm to search for a circuit that satisfies specified objectives.
- The genetic algorithm downloads a population of potential designs, coded as bit strings configurations, to the reconfigurable chip. The response of the chip in the various configurations is then measured and evaluated. Then the best bit strings configurations are transformed by genetic operators such as cross-over and mutations to create a new population of potential designs.
- The search continues using modifications of the best configurations until the termination criteria is met.

Evolvable Hardware in Electronics

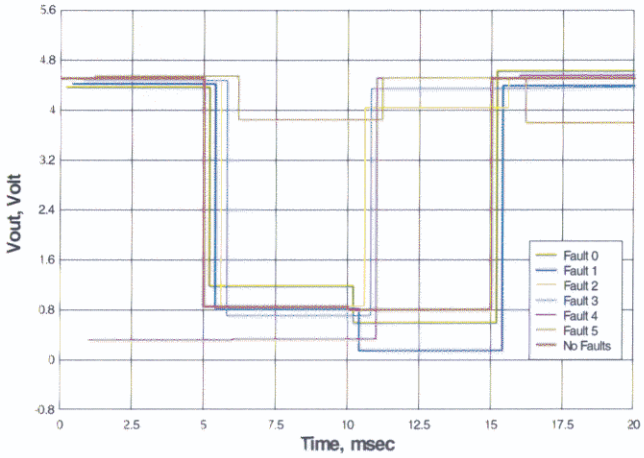
Evolutionary synthesis and adaptation of electronics circuits



Potential electronic designs/implementations compete; the best ones are slightly modified to search for even more suitable solutions

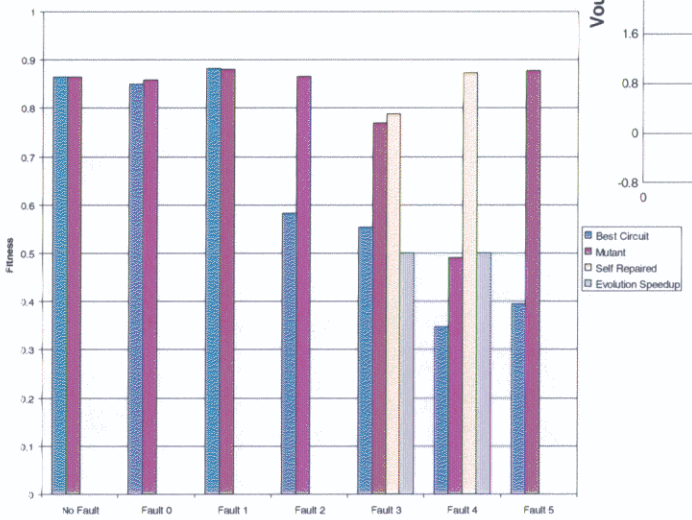
XNOR: Population based fault tolerant experiments

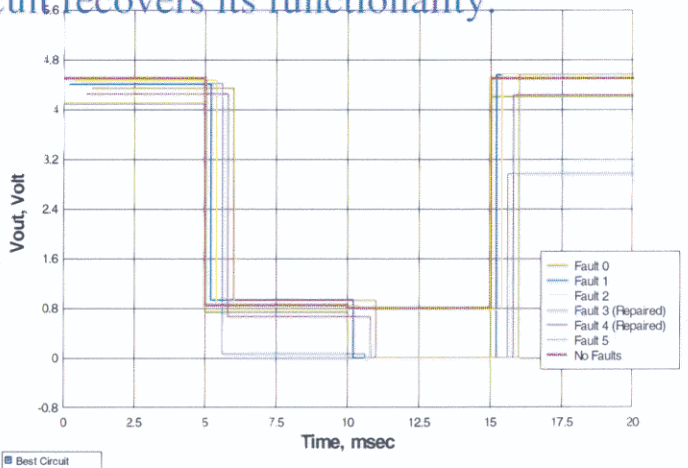
1. **Find a Particular Design**: Randomly initiate the population and start the GA. GA finds a desired circuit after 60 generations.
2. **Inject Faults**: Injected six faults, one at a time, by cutting or shorting one of the external connections between the 2 FPTA's: best circuit configuration does not achieve XNOR for faults 2, 3, 4 and 5.
3. **Two Recovery Strategies:**
   1. **Mutant strategy**: Found mutants in the population at generation 60 with better response for faults 2 and 5.
   2. **Self-healing strategy**: Restart the GA with the population at generation 60 evaluating the individuals under faults 3 and 4. After 30 generations the circuit recovers its functionality.



Best circuit response at generation 60 when 6 faults are injected

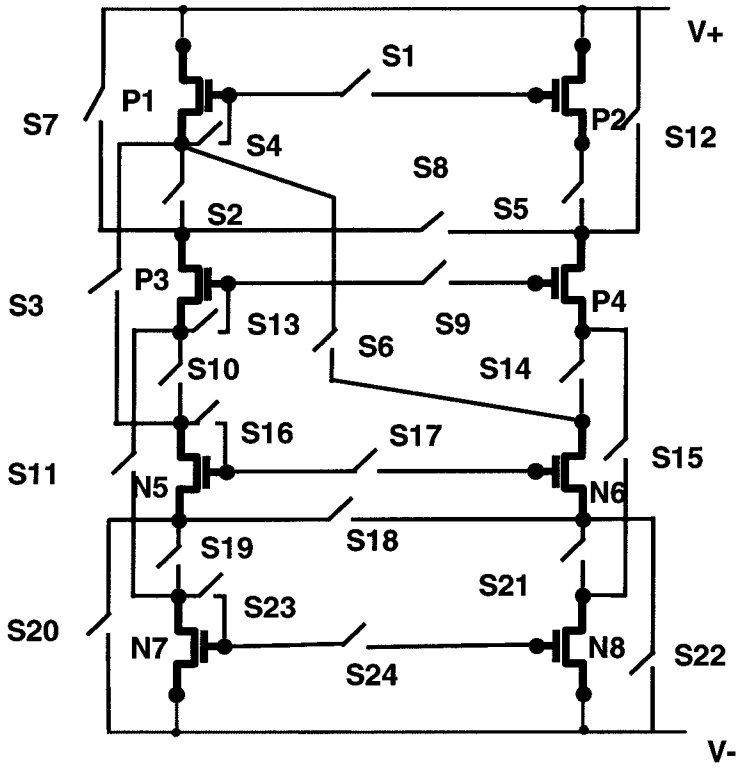
Fitness comparison

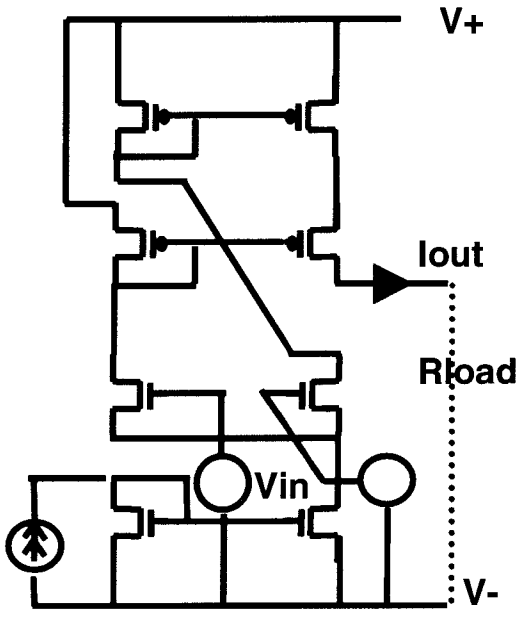




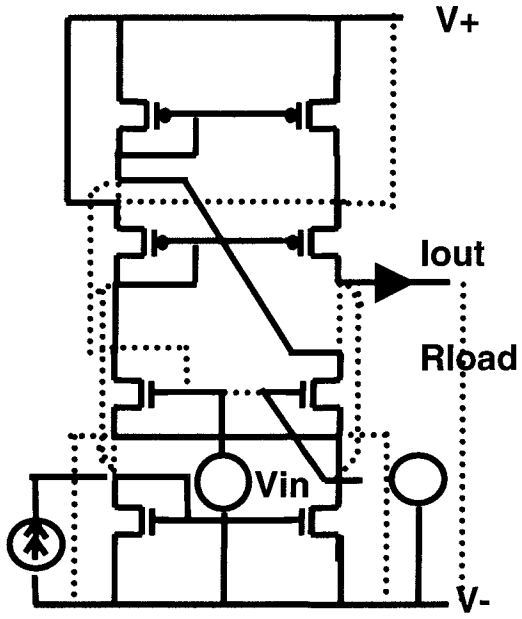
Best performing mutants for each faults (further evolution needed For faults 3 and 4

Programmable Transistor Array Cell





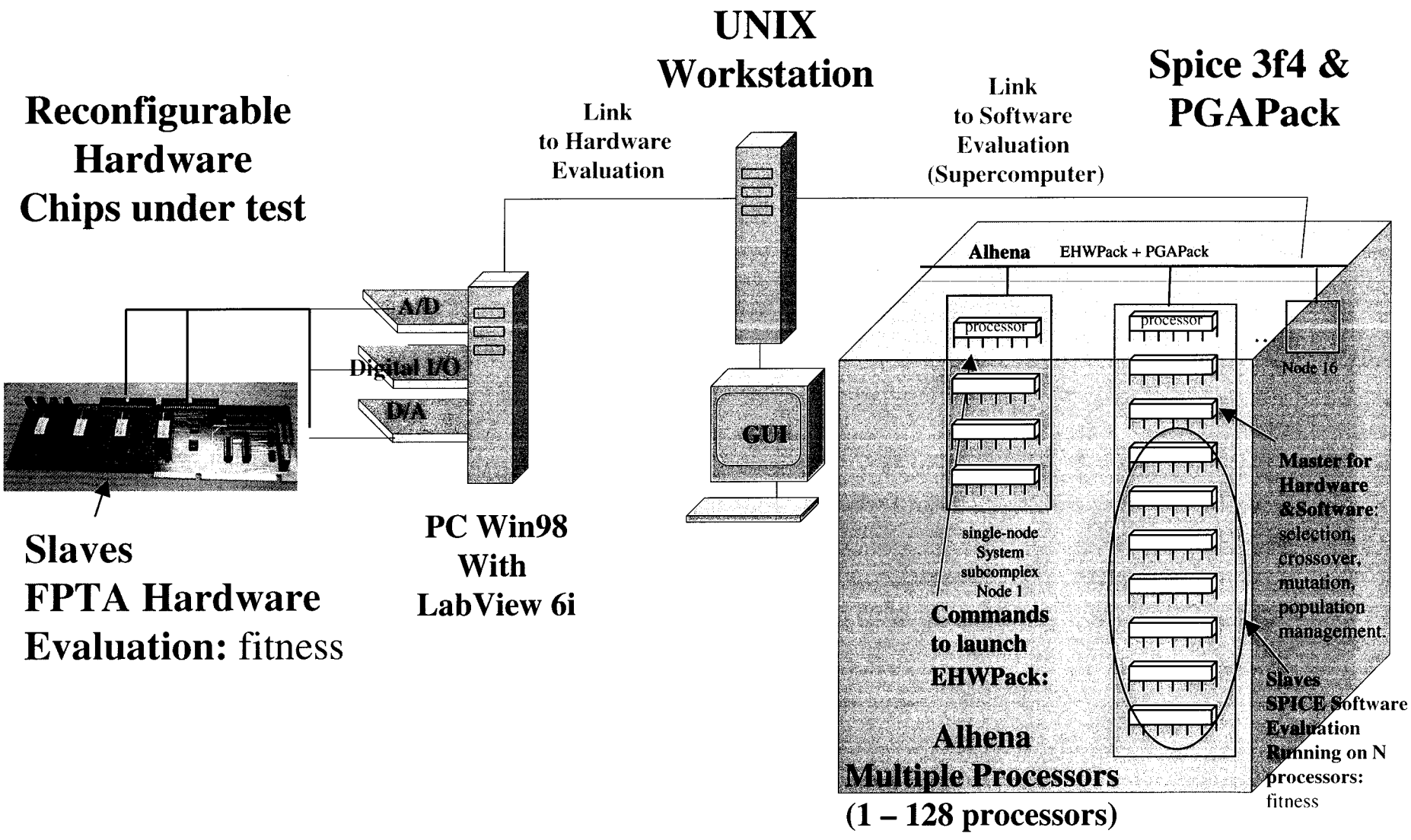
Human Design



Leakage through finite resistance OFF

- 24 programmable switches: sufficient number for meaningful topologies
- Chromosomes give the value ON-OFF of the switches
- All the terminals are connected via switches to expansion terminals
- CMOS (0.5 μ) - MOSIS

EHWPack: Testbed for Evolutionary Experiments

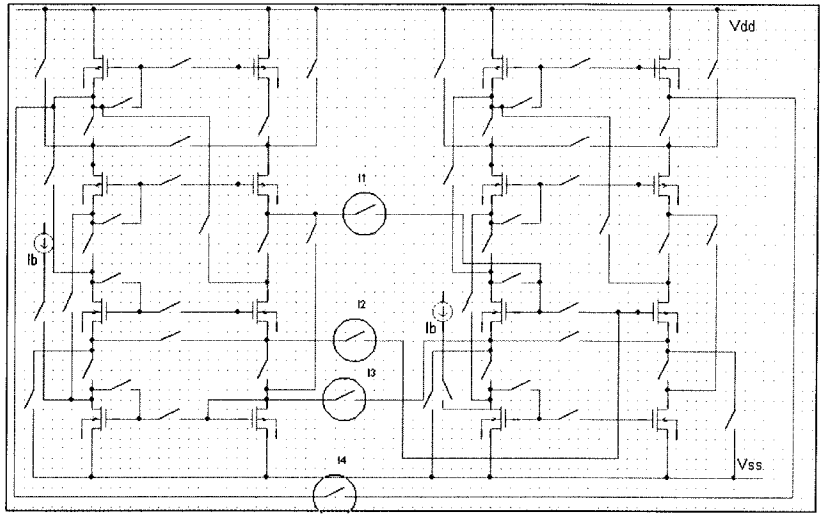


Circuits synthesized by evolution: Software

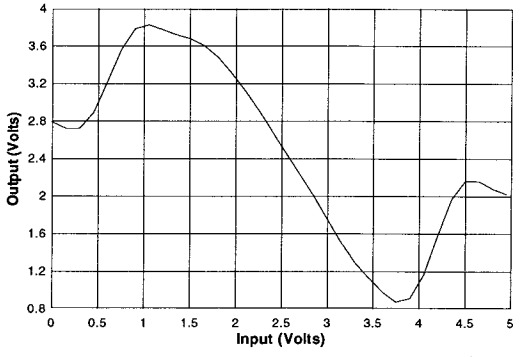
Evolved PTA Circuits:

- Gaussian: Input 0-5 Volt
  (DC Transfer on A)
- Wide band pass filter 100KHz-1MHz
  40dB/decade (AC Analysis B)
- Two-Gausian (DC Transfer)
- Multiplier (Nested DC Transfer)

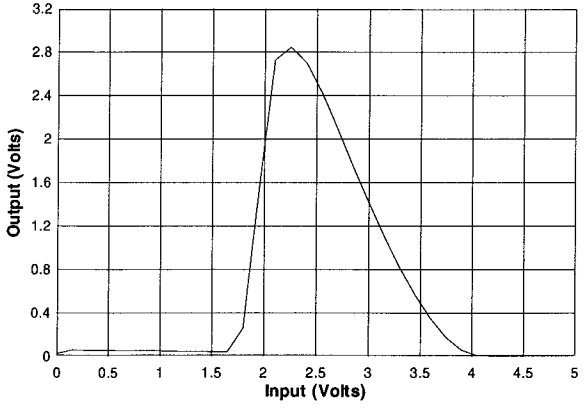
FPTA topology



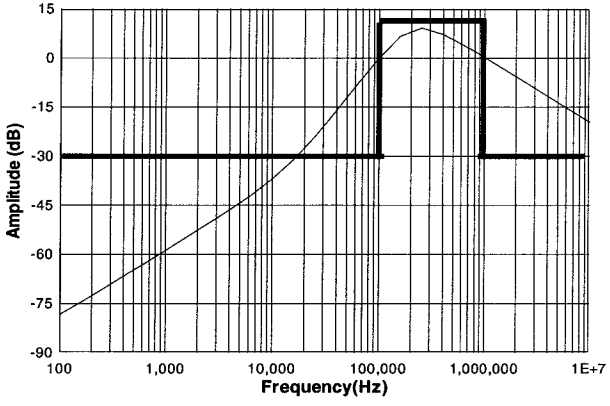
2-Gaussian

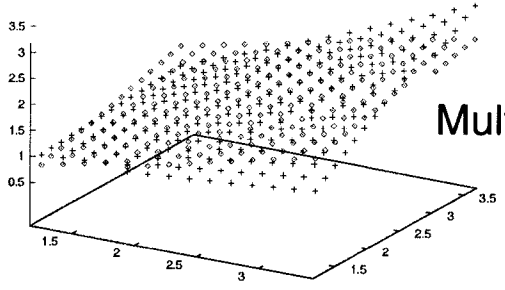


Gaussian



Filter





Multiplier

Fault tolerance by evolution

- **Evolvable hardware** (i.e. hardware that self-configures under control of adaptation/evolutionary algorithms) can preserve/recover system functionality by reconfiguration/morphing.

- **Adaptation to internal & external factors:** If the environment changes (output signal span) or the device characteristics change with temperature or radiation, one can preserve the function by finding a different circuit solution, which exploits the altered/modified characteristics.

Fault tolerant EHW: Design and Evaluation

- **Design Principles for fault-tolerant systems:**
  - **redundancy**: large number of transistors and connections able to operate in the place of the failed component.
  - **on-line (self) repaired**: swapping circuit configuration to maintain the functionality (1) by searching mutant circuit configurations in the population or (2) by running the GA during a limited number of generation to obtain a new adapted circuit configurations.

- **Evaluation of fault-tolerant systems:**
  - **Reliability**: measures how long can the system operate before malfunctioning even in the presence of faulty components. For EHW, we evaluate the performance of the electronic device when one fault is injected.
  - **Availability**: measures the proportion of the time that the system will be available for use. For EHW, we calculate the time needed by the evolution to retrieve a satisfactory circuit design.

Evolutionary Strategies for fault tolerant EHW

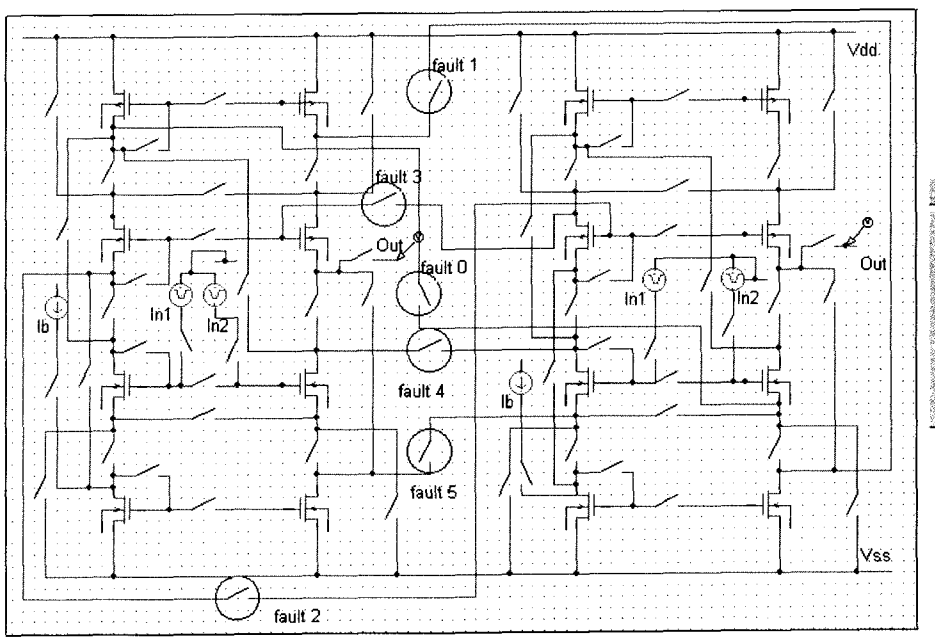
- **Fitness based fault-tolerant** (robust design):
  - Obtained the best circuit configuration by evolutionary process of a circuit with faults known a-priori.
  - Evaluate the circuit with the unknown faults.

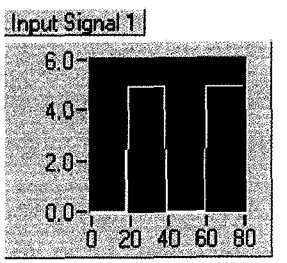
- **Population based fault-tolerant** (adaptive design):
  - Obtained the best circuit configuration by evolutionary process of a circuit without faults.
  - Extract from the population of evolved circuits, the individual which adequately performs the desired functionality in the presence of a fault (**mutant**).
  - Eventually continue the evolution to attain a performance equal to that before the fault occurred (**self-healing**).

- **Demonstration:**
  - XNOR logical function with six single faults using 2 FPTAs
  - Multiplier analog function with six single faults using 2 FPTAs
  - Gaussian: Tolerance to Faults (Movie) & Temperature

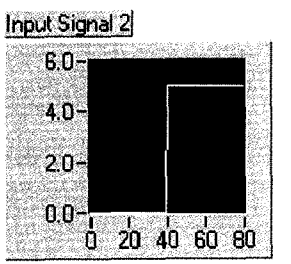
XNOR: Fault tolerant experiments

- XNOR with two cascaded FPTAs (62 bits)
  - 24 internal switches for each FPTA
  - 6 external connections between 2 PTAs
  - 4 external connections to 2 inputs voltage, 1 current bias, 1 output load.

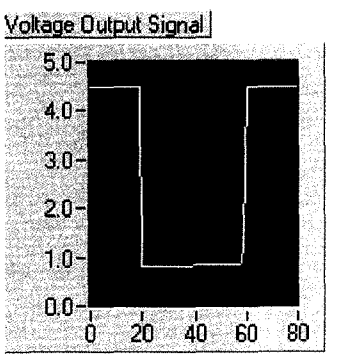




Input 1
(100 Hz)



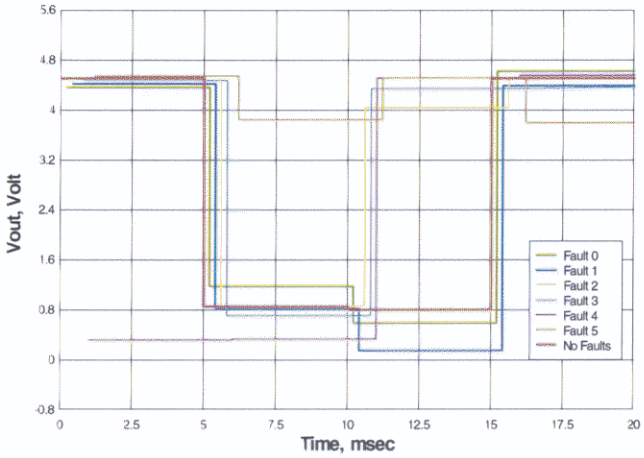
Input 2
(50 Hz)



Output

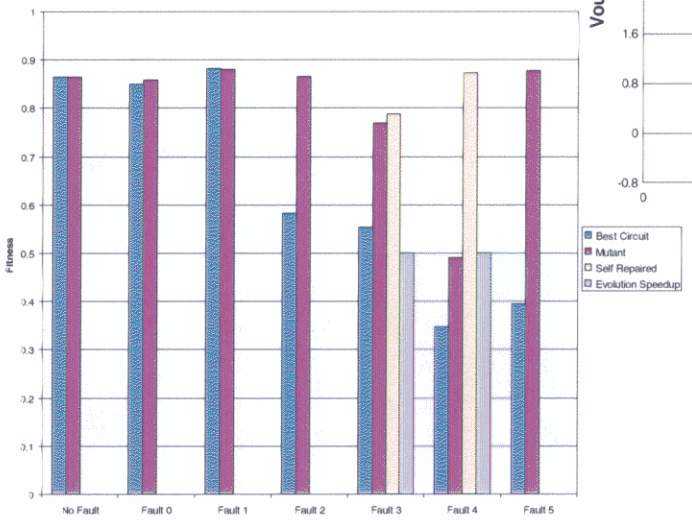
XNOR: Population based fault tolerant experiments

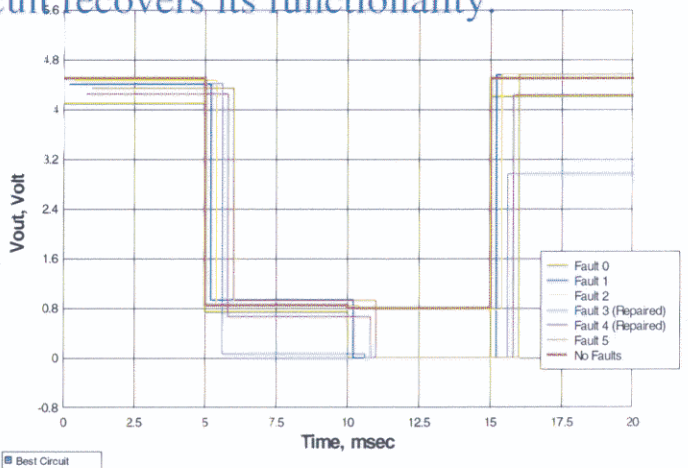
1. **Find a Particular Design**: Randomly initiate the population and start the GA. GA finds a desired circuit after 60 generations.
2. **Inject Faults**: Injected six faults, one at a time, by cutting or shorting one of the external connections between the 2 FPTA's: best circuit configuration does not achieve XNOR for faults 2, 3, 4 and 5.
3. **Two Recovery Strategies:**
   1. **Mutant strategy**: Found mutants in the population at generation 60 with better response for faults 2 and 5.
   2. **Self-healing strategy**: Restart the GA with the population at generation 60 evaluating the individuals under faults 3 and 4. After 30 generations the circuit recovers its functionality.



Best circuit response at generation 60 when 6 faults are injected

Fitness comparison

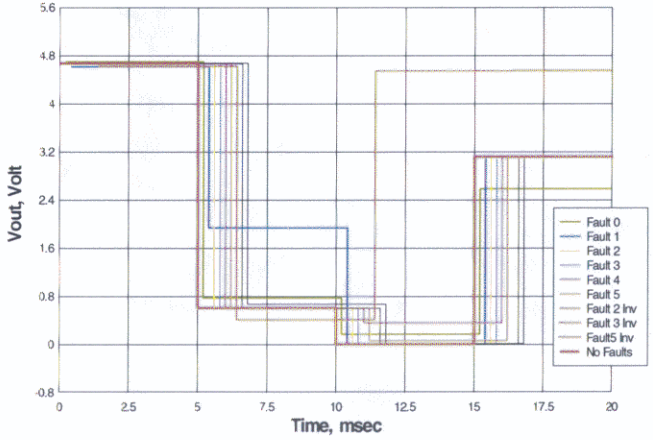




Best performing mutants for each faults (further evolution needed For faults 3 and 4

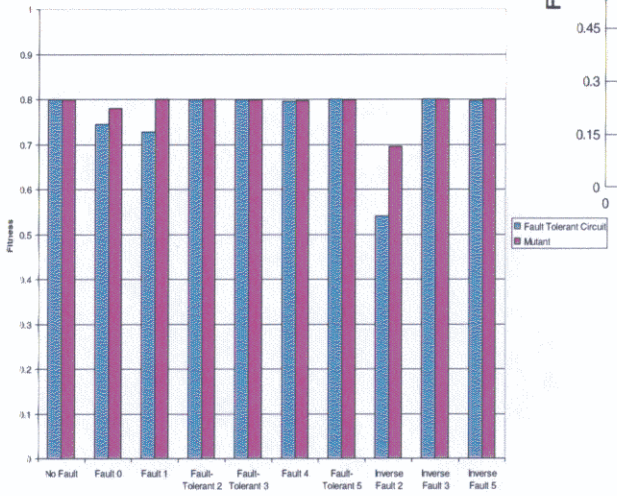
XNOR: Fitness based fault tolerant experiments

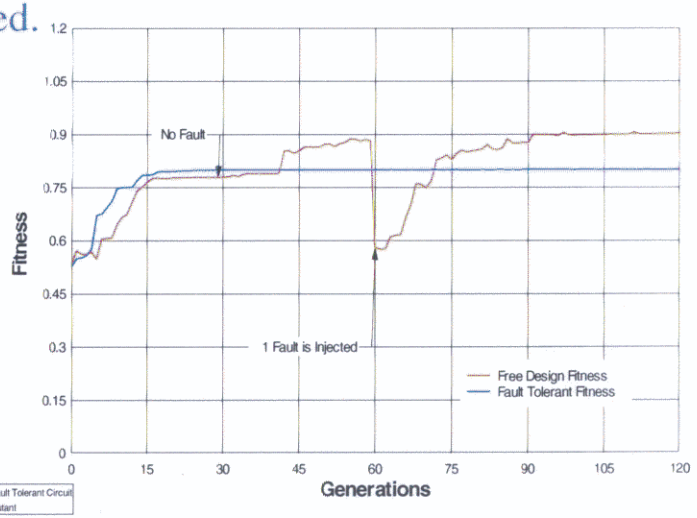
1. **Find a Robust Design**: Randomly initiate the population and start the GA. The chromosomes are evaluated in 4 different circuit states: one without faults and three with a single fault (2, 3, 5). The fitness is the average of the four evaluations. GA finds a robust circuit after 30 generations.
2. **Recovery Strategy: Fitness based**: Injected 9 faults, one at a time, by cutting or shorting one of the external connections between the 2 FPTA's. Best circuit configuration achieves XNOR
   - for faults included explicitly into the fitness function: 2, 3 and 5.
   - for faults not included into the fitness function: 0, 1 and 4 but with a low performance
   - Except when the fault injected explicitly in the fitness is inverted.



Best circuit response at generation 30 when 9 faults are injected one at a time.

Fitness comparison

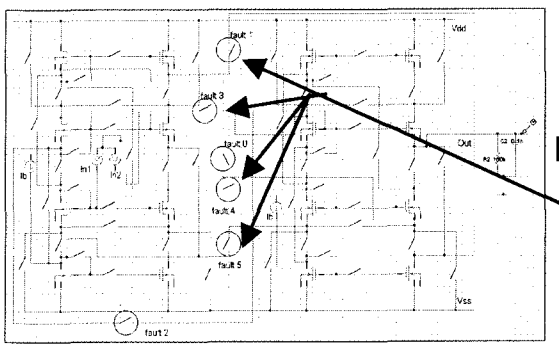




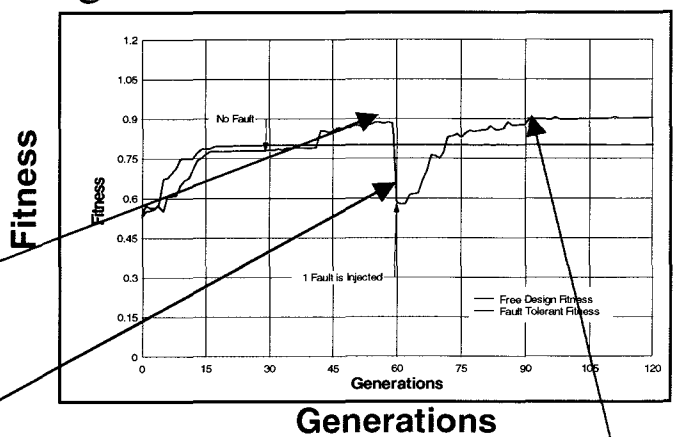
Fitness through generation for population based and fitness based

Multiplier: Self-healing Experiment for faults

- Multiplier with two cascaded FPTAs (88 bits)
  - 6 external connections between 2 PTAs
  - Find solution after 59 generations
  - Cut 1 connection after 60 generations.
  - Start the GA and recover a desired circuit after 20 generations

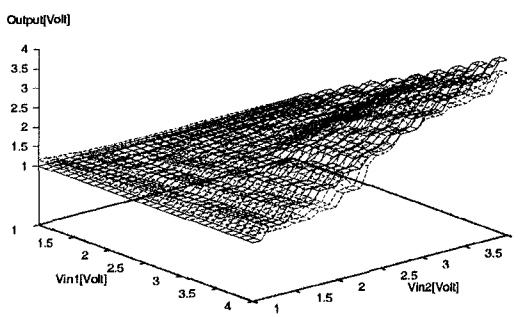


FPTA design for Fault-tolerant multiplier with 6 injected faults

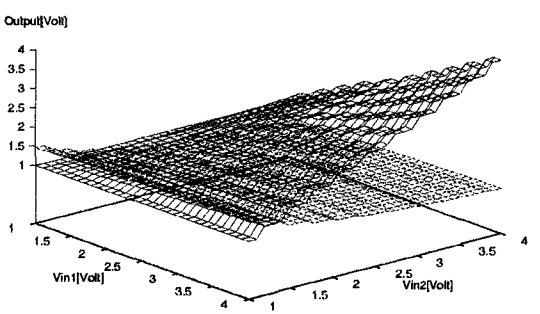


Generations

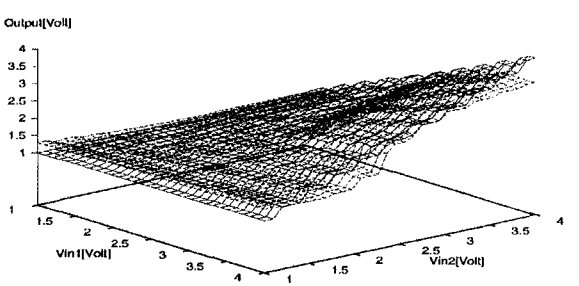
Best individual at generation 59



Fault injected at generation 60



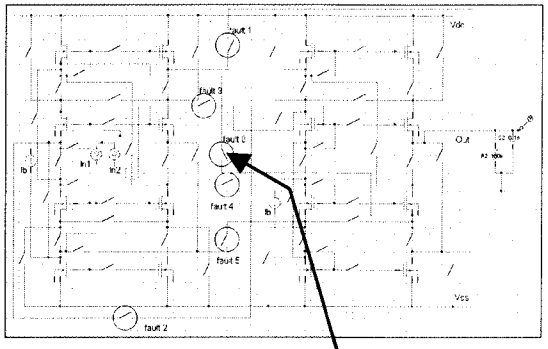
Self-repaired individual at generation 80



Gaussian: Self-healing Experiment for Faults

- Gaussian with two cascaded FPTAs (88 bits)
  - 5 external connections between 2 PTAs
  - Find solution after 200 generations
  - Cut 1 connection after 200 generations.
  - Start the GA and recover a desired circuit after 50 generations

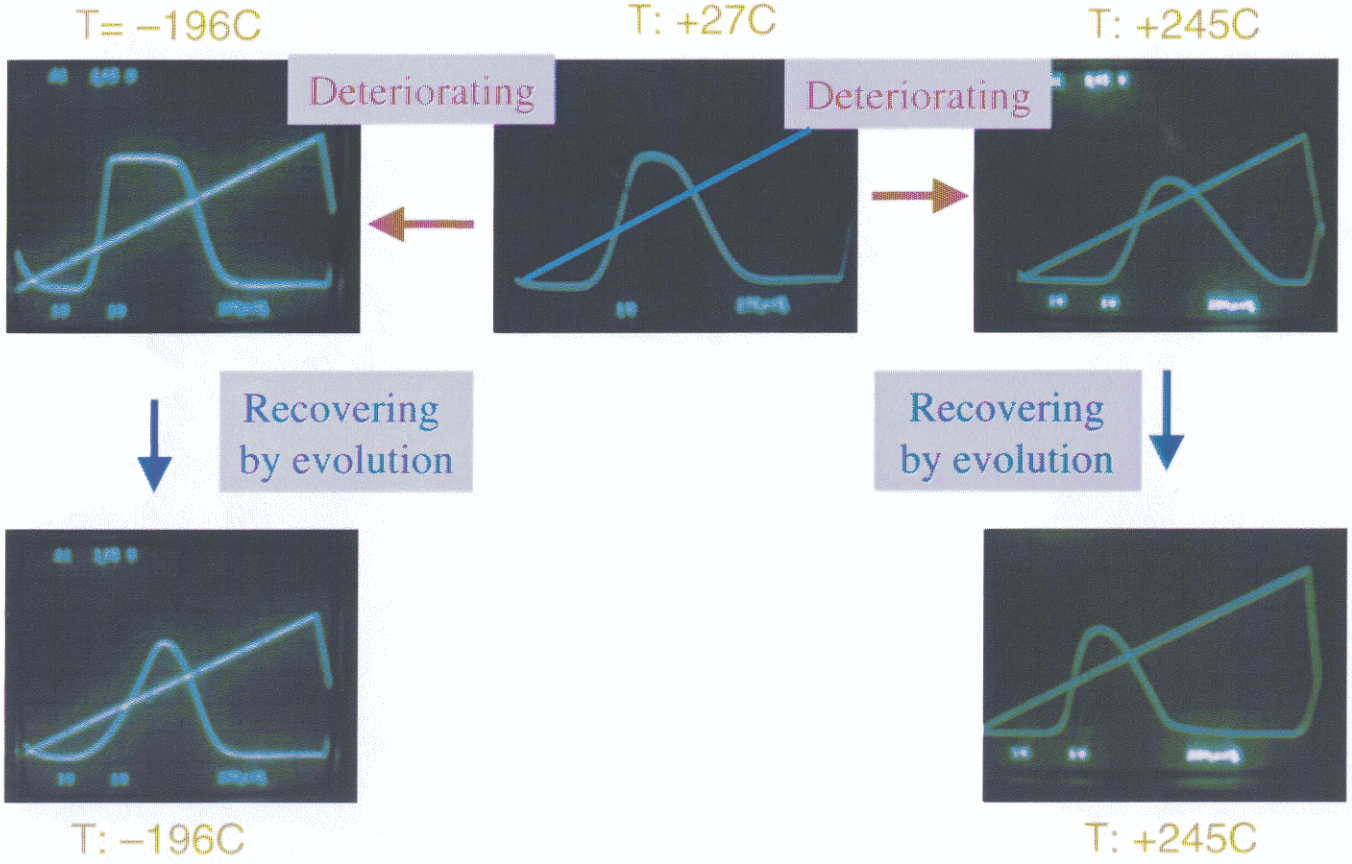
Movie: Self-healing Experiment for Gaussian circuit when 1 fault is injected



FPTA design for Fault-tolerant Gaussian with 1 injected faults

Gaussian: Self-healing Experiment for Temperature

1. Get human design or evolutionary design of a circuit at 27 C (Gaussian, NAND, AND)
2. Expose chip to low/high temperature and observe degraded response
3. Apply evolution, and obtain a new circuit solution, which recovers functionality



Immersing the chip under test in liquid nitrogen

Conclusion

These first experiments demonstrate two features enabled by evolvable hardware and which may play an important role in flexibility and survivability of future space hardware. These features are

(1) automatic synthesis of circuits to perform new functions and

(2) self-healing – recovery from faults and extreme environment.